

```

105: Account1(){
110:     P1();
115:     Status = P2(Temp);
120:     if (Status==ERROR1){
125:         do-reverse-of-P2();
130:         do-reverse-of-P1();
135:         exit ();
138:     }
139:     temp-was-one=0;
140:     If (Temp==1){
145:         P3();
148:         P4();
150:         P5();
152:         temp-was-one=1;
153:     }

155:     Status=P6();
160:     if (status==ERROR2) {
165:         do-reverse-of-P6();
170:         If (temp-was-one==1){
175:             do-reverse-of-P5();
178:             do-reverse-of-P4();
180:             do-reverse-of-P3();
183:         }
185:         do-reverse-of-P2();
190:         do-reverse-of-P1();
195:         exit ();
197:     }
199: }

```

FIG. 1
(Prior Art)

```

205: Account10
206: {
207:     Typedef int (*fptr)();
208:     Tstep(int, fptr, fptr);
209:     int txid = Tinit();

215:         Tstep (txid, P1(), R1());

225:         Status = Tstep (txid, P2(Temp), R2());
230:         if (Status == ERROR) {
240:             Tabort (txid);
245:             exit();
247:         }

250:         if (Temp==1) {
255:             Tstep (txid, P3(), R3());
260:             Tstep (txid, P4(), R4());
265:             Tstep (txid, P5(), R5());
267:         }

270:         Status = Tstep (txid, P6(), R6());

275:         if (Status == ERROR) {
280:             Tabort (txid);
285:             exit();
290:         }

295:         Tcommit (txid)
299:     }

```

FIG. 2

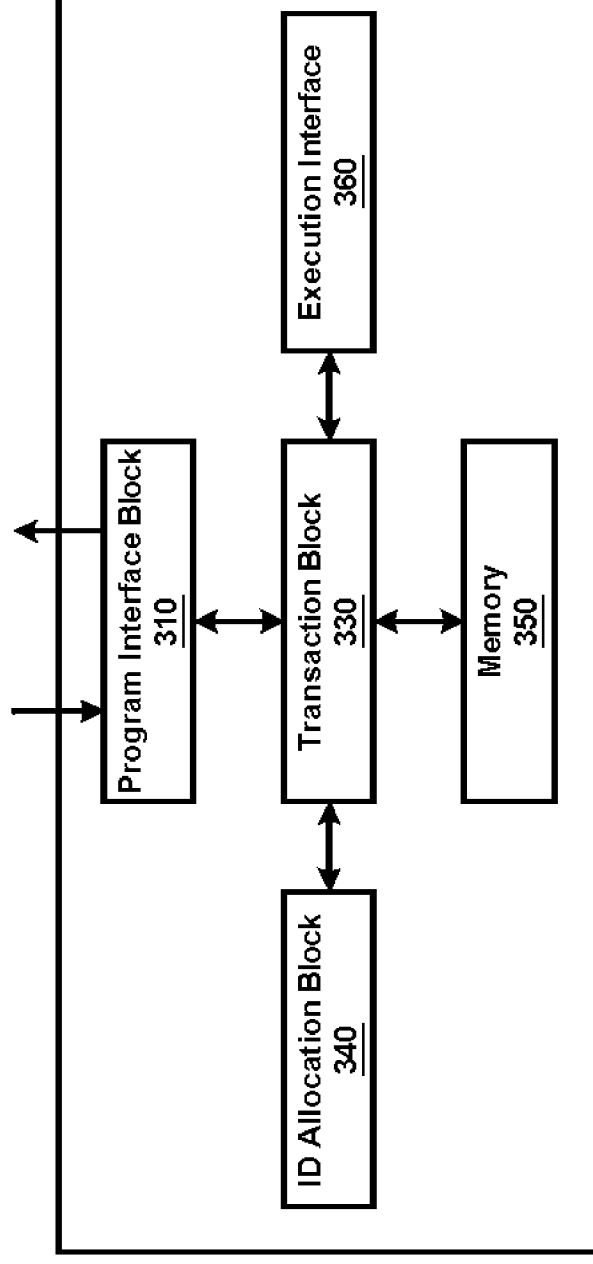


FIG. 3

Transaction Id	Task Procedure	Roll-back Procedure	Stack Status
200	P10	R10	R10
200	P20	R20	R20, R10
300	P10	R10	R10
300	P20	R20	R20, R10
300	P30	R30	R30, R20, R10
300	P40	R40	R40, R30, R20, R10
300	P50	R50	R50, R40, R30, R20, R10
200	P60	R60	R60, R20, R10
300	P60	R60	R60, R50, R40, R30, R20, R10

FIG. 4

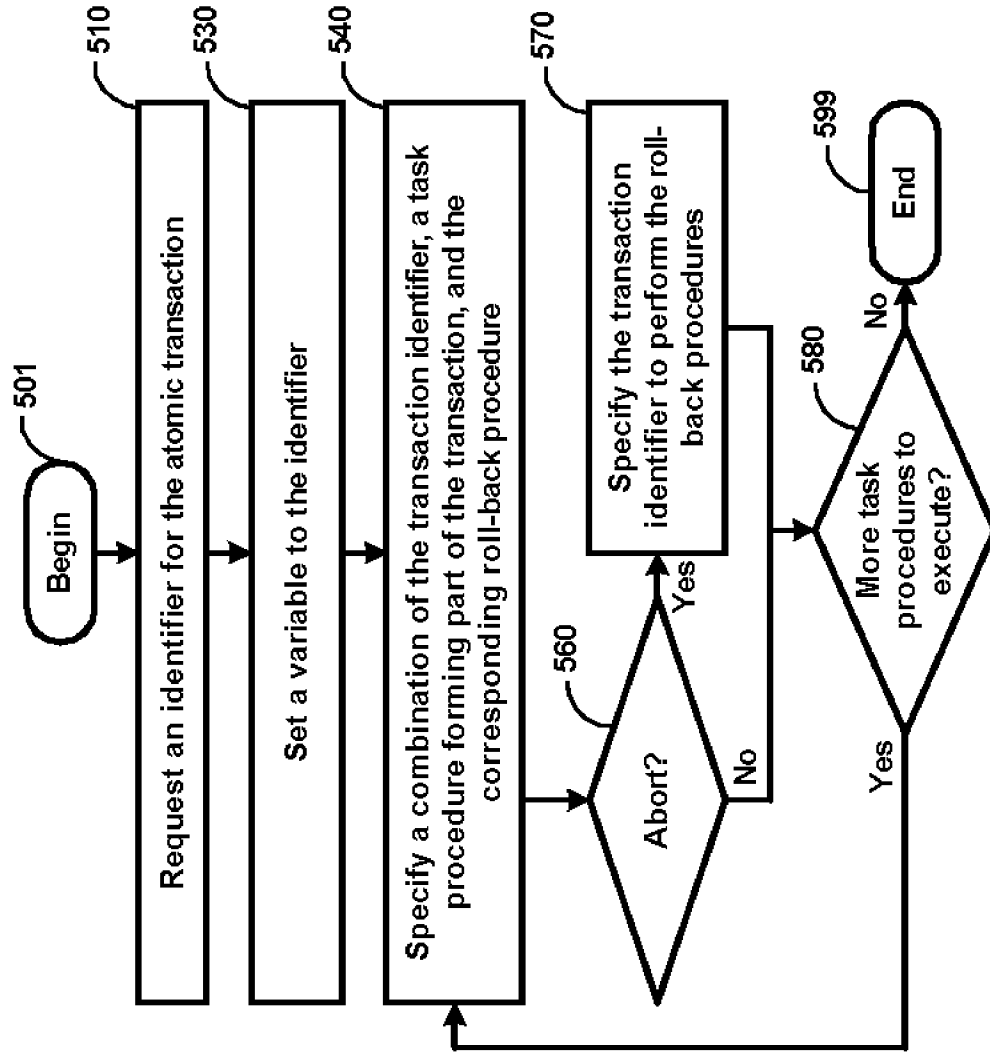


FIG. 5

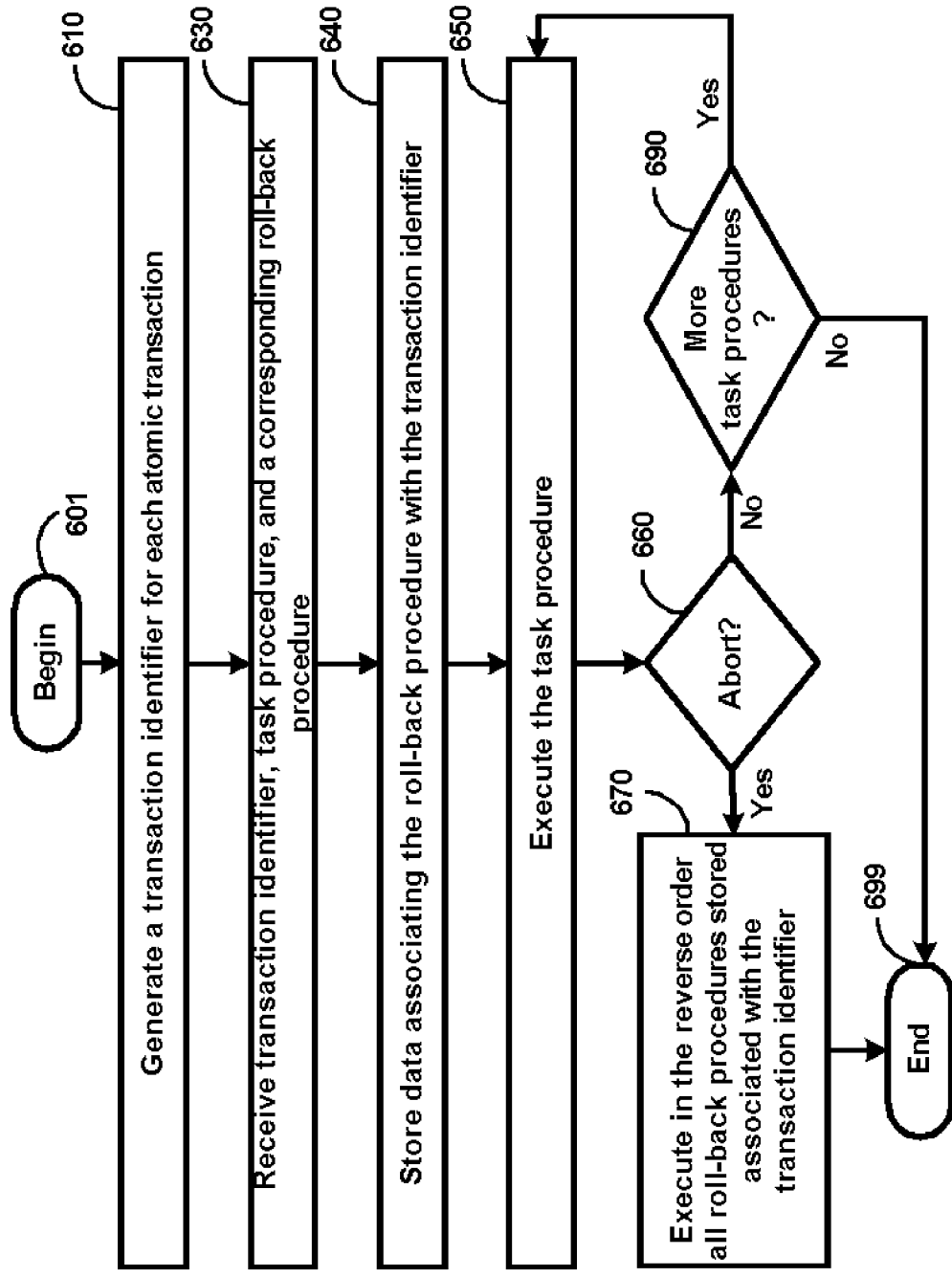


FIG. 6

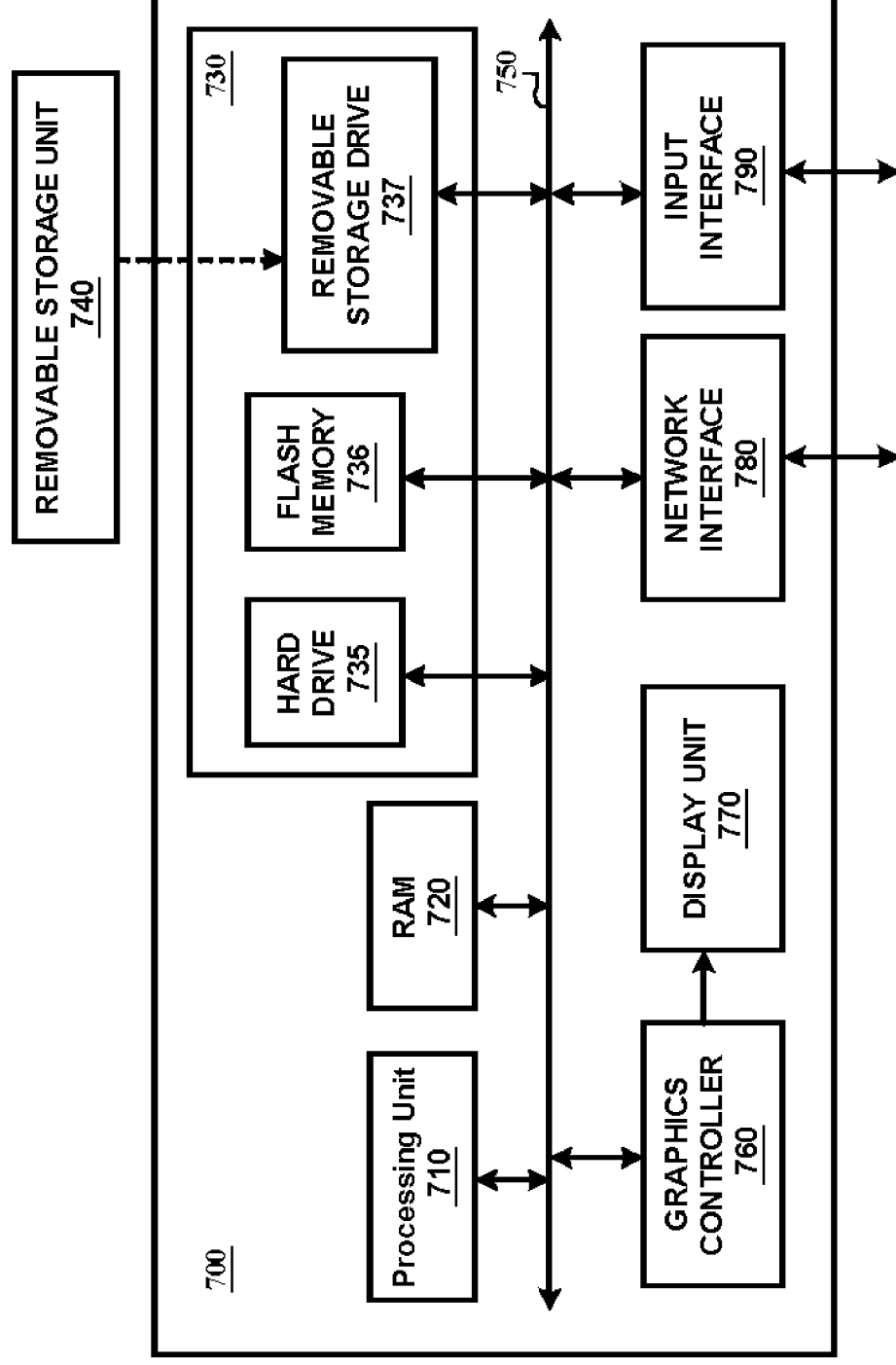


FIG. 7